Web Service Discovery Research and Implementation Based On Semantic Search Engine

Chao Ma
Beijing University of Posts and Telecommunications
Beijing P.R. China
xdhaixiao@163.com

Meina Song
Beijing University of Posts and Telecommunications
Beijing P.R. China
mnsong@gmail.com

Ke Xu
Beijing University of Posts and Telecommunications
Beijing P.R. China
permit@263.net

Xiaoqi Zhang
Beijing University of Posts and Telecommunications
Beijing P.R. China
alphazxq@gmail.com

Abstract

Common Services researched by Service Science have four major provision modes: web page mode, local client mode, web service mode and cloud service mode. Currently the most common used is Web Service mode. And now most Web Services provided by every Service Provider access and discovery strategy is still based on keyword search's UDDI. Services returned to users often can't be called. This paper will raise a new Web Service access and discovery strategy which combines with Search Engine technology and Semantic Web technology. Search engine technology provides a search capability based on lexical analysis and grammar analysis while Semantic Web technology provides a matching and scoring capability based on semantic information. Search result lists returned to users will sort from higher score to lower score. This Web Service discovery strategy will improve service's recall ratio and precision and it is very practical in engineering.

Keywords: web service; search engine; semantic web; lexical analysis; grammar analysis

1. Introduction

With more and more Web Services are developed by service providers, how to discovering and locating services accurately has become more and more important for service requestors. And existing Web Service discovery technology are essentially based on keyword match. Web Service searcher searches services according to keywords provided by users. But result services obtained by this way often do not meet users' need. With Search Engine technology and Semantic Web technology becomes more and more maturely, I combine Web Service discovery strategy with Search Engine and Semantic Web technology in this paper. Let's call this strategy as Semantic Web Service Search Engine. It will make a lexical analysis and syntax analysis to a user's request and then match target services semantically. The recall ratio and precision will improve greatly after this series of operations.

978-1-4244-6359-6/10/$26.00 ©2010 IEEE

2. Relate Work

Web Service discovery is an important component in Web Service architecture. The so-called Web Service discovery is users find out services they want among these different Web Services so as to execute their Web Service request.

In traditional Web Service register and discovery, service providers publish their service description on UDDI, and service requestors obtain service information they want by submitting query request. We can discovery Web Service on Internet, get service description and bind service dynamically through UDDI、WSDL and SOAP. Traditional Web Service discovery base on Figure1.

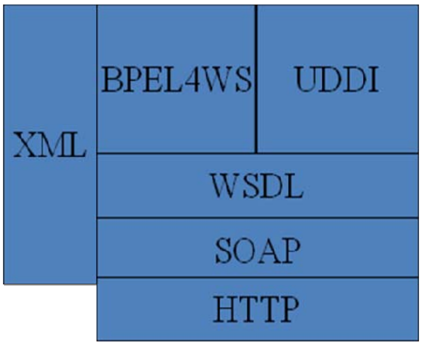


Figure1. Traditional Web Service Discovery Framework

Traditional Web Service discovery is mainly based on syntax level keyword and framework service discovery these two discovery strategy. Keyword based Web Service discovery is very similar with keyword based search engine. But this method has some defects:

It can't describe query target clearly.

It can't measure the degree of consistency between candidate and query target.

It can't use refinement, generalization, level-level semantic operations to query.

The first two points are the main factors of affecting recall ratio, while the third point is the main factor of affecting precision. Frame-based Web Service discovery technology is a improvement than keyword based Web Service discovery technology, and now most Web Service discovery technology is based on frame currently, such as WSDL/UDDI technology. UDDI is lack of semantic information support, it only supply with service basic description and frame-based matching mechanism and this makes service discovery base on keyword matching. The kind of service discovery strategy not only can't distinguish same semantic information without same grammar, but also can not distinguish same grammar information without same semantic. So it can't provide Web Service discovery based on service function. Further more It does not support service combination's characteristics, this makes the service matching accuracy is not high, matching method is not flexible enough. So it can't meet the requirement of automatic discovery and composition.

3. Search Engine Full Search

Search Engine is defined as searching information on the internet using the specific program with some strategy, after organization and process to the information, return it to users. Its responsibility is providing search service to users [1].

This section discusses several modules of a Search Engine System as well as the interaction between each other.

A Search Engine System consists of Spider, Document Parser, Analyzer, Indexer, Searcher, Sort, Filter and UI as the Fiture2 showed below:

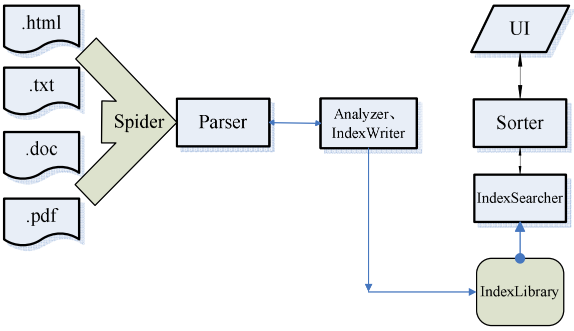


Figure2. Search Engine Structure

3.1Spider

Spider's responsibility is to fetch web pages. It collects information on the Internet day and night and then sends the information to the Search Engine's database. A spider is commonly implemented with distributed and parallel computing skill.[2]The amount of information discovered by a Business Search Engine can reach as more as millions web pages.

Spider visits a single web page of the start set firstly, extracts the valuable imformation of the web page, and then sends it to the Search Engine's database. The spider then jump to another web page and it repeats this job. Users can submit their requests if they want to let web pages created by themselves be taken by the Search Engine.

During the development of a Search Engine, fetching needed web pages' information with a proper spider is just the first step, but this step is quite important, because actually Search Engine is a huge resource of which the primary premise is solving users' need from the resource.

3.2 Document Parser

Original documents fetched by Spider automatically will be parsed by Document Analyzer, and its main function is filtering file information so as to provide a property way for index output [3]. Firstly we should filter information, separating HTML、JSP、ASP etc from large amount of labels which are used to describe documents format. Then lexical and syntax analysis will be followed.

3.3 Analyzer

Analyzer segments document resource parsed by Document Parser. It will segment text information into the smallest unit which can be indexed according to certain rules.

Indexing and searching both need analyzers. In order to obtain results correctly, during the processes of indexing and searching must use the same analyzer. A analyzer process can be illustrated as Figure3 showed:

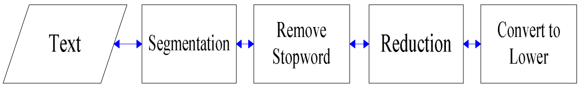


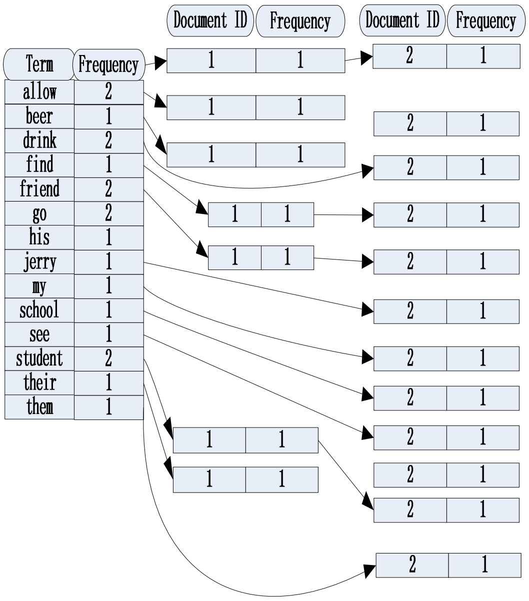
Figure3. English Analyzer Process

Reduction is removing morphological changes of word endings so as to convert these words to their prototype. For example worked->work. The so-called stop word is that some word's frequency is very high, but the information carried by them affect the context nearly. For example "a, an, the" and all kinds of punctuations. Stop words will be filtered after analyzed, they won't be indexed.

3.4 Indexer

We need a method to preprocess documents so as to build a data structure easy to be searched between documents. This kind of data structure is index. Indexing will significantly improve speed of the information search when Search Engine needs to deal with large amount of documents.

Actually there are three kind of indexing, inverted index, suffix arrays and signature of document. Inverted index has been widely used by most information system currently. In theory, the inverted index is a mechanism for word. Usually, it consists of keyword and occurrence. Every keyword which has been indexed will be followed a list used to this word's position in all documents. The following Figure4 shows an example of inverted index:



Term	Frequency
allow	2
beer	1
drink	2
find	1
friend	2
go	2
his	1
jerry	1
my	1
school	1
see	1
student	2
their	1
them	1

Figure4. Inverted Index Structure

3.5 Index Searcher

Index searcher obtain users' search request from UI, search with certain strategy in the index library and then pass search results to sorting and filtering system. The procedure is as followed:

User input query words

Process the query words with lexical analysis、syntax analysis and language process. Lexical analysis is mainly used to identify words and keywords; syntax analysis is mainly to build a syntax tree based on syntax rules; language process is nearly the same with language process during indexing process, such as convert learned to learn.

Searching the index so as to obtain documents with syntax tree above.

3.6 Sorter And Filter

Documents searched by Index searcher are not enough, we need sort query results In accordance with the relevance of query, the more relevance the more front. We see query statement as a short document, scoring for the relevance between documents, and higher score will be displayed front than lower score.

Firstly a document is consists of many terms, such as Lucene, full-text, this, a, what etc. Secondly different terms' importance is different among documents, for example search, Lucene, full-text is more important in this article, and this, a, what may not be so important. So if two documents both contain search, Lucene, full text, these two documents' relevance are better. But even if a document contains this, a, what, another one document does not contain this, a, what, does not affect the relevance of these two documents. Thus determine the relationship between two documents, first we should find out what words are important to documents. Finding out what words are important to documents is called calculating Term Weight process. There are two parameters during calculating Term Weight, and one is term while the other is document. Term Weight represents this word's importance in the document, and the more Term Weight is the more important the word is. Thus they will play more important rule in calculating documents' relevance. We use a algorithm called Vector Space Model to determining relationships among terms so as to calculating documents' relevance.

3.7 UI

This module is responsible for human-computer interaction functions, receiving queries from the user, distributing the query to Index searcher and display results which have been sorted and filtered to users.

4. Semantic Web

4.1 Concept

In the semantic network, information is given explicit meaning, and then machines can process and integrate useful information on the Internet. Semantic web use xml to define tag format and use RDF to express data. Simply says, semantic web is a intelligent web who can understand human language. It not also can understand human language, but also can make computers' communication as easy as human's communication.

Not likes existing internet, semantic web's data is supply for people's use. The new gyration WWW will also be able to provide data which can be processed by computers, this makes a large number of intelligent service possible。

4.2 Implement

Berners-Lee proposed the architecture of the Semantic Web in 2000 and described it simply. This architecture has seven layers whose functions gradually increase from bottom to top.

The first layer: Unicode and URI. Unicode is a character set. Every character in this set is represented by two bytes. It can represent 65536 characters, and it nearly contain all language words in the world.

The second layer: XML + NS +xml schema. XML is a simplified SGML. It combines SMGL's rich functionality and HTML's easy of use. It allows users to add any structure. NS which stands fro naming space whose purpose is to avoid different applications use the same characters to describe different things is decided by URI index. XML schema is a representation for DTD, it is described by XML grammar, but it's more flexible than DTD. It provides more data types and can server better for XML documents.

The third layer: RDF + rdf schema. RDF whose goal is create a a variety of metadata's co-existence's framework is a kind of language who can describe information on WWW. Rdf schema use a system that can be understood by machines to define resource vocabulary. It's aimed to provide a mechanism or framework for vocabulary embedding. Under this framework many kinds of vocabulary can integrate together to describe web resource.

The fourth layer: Ontology vocabulary. This layer is in concept and relationship's abstract description base on RDF. It is used to describe the applications of knowledge, describe various types of resources and the relationship among them. and implement the expansion of vocabulary. In this layer, users can define not only the concept but also define rich relationship among different concepts.

The fifth to seventh layer: Logic, Proof, Trust. Logic is responsible for providing axioms and inference rules. Logic once established, it can be logical resources, the relationship between resources and the reasoning results have proved their effectiveness. Proof, as well as through the exchange of digital signatures, establish a trust relationship, which proves the reliability of the Semantic Web's output and whether it meets users' requirements.

5. Prototype

As various limitations of traditional Web Service discovery strategy, services which are listed to service requestor after then send service request have a very low recall ratio and precision. Service developed by service provider are quite difficult to register into UDDI and developed by users. So improving Web Service discovery strategy is a very important issue in Service Science all the time. The section will combine the search engine technology described in the second section and the semantic web technology described in the third section to improve the traditional Web Service discovery technology. It has improved service's recall ratio and precision to some extent.

5.1 Service Description

Semantic description is an important step in Web Service discovery. OWL-S which stands for Ontology Web Language for Services provides the semantic foundation for Web Service publish and request.[4] It helps service requestors to discover services they really need. OWL-S consists of three parts:

Service Profile: it describes service's content and it's used to service publish and discovery.

Service Process Model: It describes services how to work, such as execution logic order.

Service Grounding: It describes how to access services, such as communication protocol and other specific details.

5.2 Service registration

Service provider will publish the service after he has developed it and described it in OWL-S. Service provider sends a service registration request to the service register and then sends this service's OWL-S to it. The service register will extract two kinds of information: the first one is traditional WSDL address and the second one is service semantic information. This step is a improvement than traditional Web Service discovery. The step is as followed:

Analyze and index keywords of service description. Use semantic information base illustrated in 4.3 sections as the analyzer dictionary. And then the engine corresponds services to the appropriate types of service.

Extract input and output parameters and then generate random values to call the service so as to calculate this service's concurrency and throughput. Put these values into the service's semantic information base.

Expand the semantic information base according to service's description.

Score the service according to the information above, and then combine it with services have been resisted. After this procedure, we may get some new services.

5.3 Semantic Information Base

Semantic Web Service is based on a set of rules and ontology [5]. What's called ontology is a set of expanding concept. Semantic information base is the set of expanding information described above. As more and more Web Service registered into registration center, more and more new concept will be found and merged into semantic information base. And then corresponding inference rule will be applied to this new concept and new Web Service category as well as combination will generate.

5.4 Semantic Web Service Search Engine

Semantic Web Service Search Engine is this paper's core content. Its running process is as followed:

1) Use search engine's lexical analysis and syntax analysis to user's query request.
2) Use search engine's analyzer to user's query request. Filter services in service base on service name and service type to remove different type with request service. This paper use service Category in OWL2S Profile specification to describe Web Services classification information, that is Web Service type and use ontology concept in OWL to tag service's service Category information. [6]By calculating similarity between request services and target services, it will return services whose similarity is greater than given threshold.
3) Through above two points we can get service list based on keyword matching, but these services are not accurate enough to return users as their input output parameters and quality may not meet users' need. So this needs further process below.
4) Match services in result service list according to users' input and output parameters. We score the services using vector space method. Set up a dimension coordinate system with input similarity as a dimension output similarity as a dimension and service description as a dimension. Finally services with high weight will have the priority of returning firstly. Services' recall ratio and precision returned by this strategy have greatly improved.

Services input similarity and output similarity are illustrated as below:

(1) Input Similarity

Let's suppose target service has m input parameters wsin=(Ip1, Ip2, ,Ipm), request service ws_r has n input service ws_rin = (Iq1 , Iq2 , ,Iqn), function sim(Ipi, Iqj). Calculate the similarity between input parameter Ipi in wsin and input parameter Iqj in ws_rin. Input parameter similarity calculates as followed:

simin (wsin ,ws_rin) = (maxΣ mi = 1 Σnj = 1sim (Ipi , Iqj) 3 xij) /min (m, n)(AP) s. t. Σnj = 1xij = 1 i = 1, 2, ,mΣ ni =1xij = 1 j = 1, 2, , n. The first constrant means that the ith parameter Ipi in wsin can only match a parameter in ws_rin once. And the second constraint means that the jth parameter Iqj in ws_rin can only match a parameter in wsin once.

(2) Output similarity

Output similarity's formula simout (wsout , ws_rout) is similar with input similarity's formula.In short, calculating input similarity and output similarity both match parameters between two services and calculate it similarity, see the maximum of seminary's sum as input and output similarity.

5.5 Service Searcher

This module is Web Service discovery system's UI, it is responsible for receiving users' requests and dispatching them to Web Service Search Engine. System will send the result lists to users after it discovers target services.

6. Conclusion

As traditional Web Service discovery technology is only based on keyword matching, recall ratio and precision is very low, this paper combines traditional Web Service discovery strategy with search engine technology and semantic web technology. It makes syntax level keyword matching more precisely and add semantic information to services additionally. Final service result list is weighted by keyword matching and service semantic vector, so it has improved service recall ration and precision greatly.

7. Acknowledge

This work is supported by the National Key project of Scientific and Technical Supporting Programs of China under Grant Nos.2008BAH24B04, 2006BAH02A03; the Program for New Century Excellent Talents in University No.NCET-08-0738; and the Innovation Technology Star Program of Beijing under Grant No.2007A045.

8. References

[1]Web information resource retrieval and use. Beijing: Tsinghua University Press, 2005. [M].

[2] Semantic Search Engine Introduction. [EB/ OL] . [2008 - 09 -22] . http :/ / xchuspace. spaces. live. com/ blog/ cns ! 3f3f394f3a76da53 ! 115. entry , accessed in Oct. 11 ,2005

[3] Norbert Lossau. Search Engine Technology and Dig2 ital Libraries[J] . D - Lib Magazine. 2004 , (6)

[4] Massimo P, Julien S,Nsvrrn S.A Broker forOWL2SWeb Services[C].In Proceedings of the AAAI Spring Symposium, Palo Alto, California,2004

[5] Terry R,Massimp P, Katia S. Advertising and matching DAML2S service descriptions[C]. In: Proceedings of the International SemanticWeb Working Symposium,Amsterdam: IOS Press, 2001: 4112430.

[6] Klein M, Bernstein A. Searching services on the semantic Web using process ontologies [C]. In: Proceedings of the International Semantic WebWorking Symposium,Amster2dam: IOS Press, 2001: 1592172.